

[Title]

Method and device for controlling a queue buffer

5 [Field of the invention]

The present invention relates to a method and device for controlling a queue buffer, where said queue buffer queues data units in one or more queues.

10

[Background of the invention]

In data unit based communication, i.e. in which an information to be transmitted is divided into a plurality of units, and the individual units are sent over a communication network, it is known to provide queue buffers at links along the network, such that data units transported over such a link may be buffered. The buffer may be a sending or output buffer (i.e. a buffer for data units that are to be sent over the outgoing link) or a receiving or input buffer (i.e. a buffer for data units that have been sent over the incoming link).

Such units for transporting data may carry a variety of names, such as protocol data units, frames, packets, segments, cells, etc., depending on the specific context, the specific protocol used and certain other conventions. In the context of the present document, all such units of data shall generically be referred to as data units.

30

The procedures for placing data units into a queue, advancing them in the queue, and removing data units from the queue are referred to as queue management or queue control.

35 Many concepts of queue management are known, including such concepts in which a queue length parameter (e.g. the absolute length of queue or an average length) is monitored and

compared with a length threshold value, in order to perform a congestion notification procedure if the threshold value is exceeded. Such a congestion notification procedure can consist in dropping a data unit and/or in marking data units (e.g. setting a congestion notification flag in data units). Various examples for such management concepts, like drop-on-full or random early detection (RED) are described in the introduction of EP-A-1 249 972. As a consequence, a repetition is not necessary here.

EP-A-1 249 972 proposes a scheme in which the length threshold value that is to be compared with the queue length parameter is automatically updated on the basis of one or more characteristics of the link over which the queued data units are to be sent. By adapting the automatic threshold to one or more link characteristics, a very flexible form of active queue management is obtained that may provide improved throughput and reduced delay, especially over links that have time varying characteristics such as wireless links. Particularly, EP-A- 1 249 972 suggests estimating a link capacity value based on the link's data rate and adapting the threshold value on the basis of the estimated link capacity.

[Object of the invention]

It is the object of the present invention to provide an improved method and device of queue buffer management.

[Summary of the invention]

This object is achieved by a method and device as described in the independent claims of the present application. Advantageous embodiments are described in the dependent claims.

In accordance with the present invention, in a system where queue buffer control or queue buffer management comprises

determining a value of a length parameter related to the queue length, comparing this value with a threshold and performing a congestion notification procedure if the threshold is exceeded, and in which an automatic threshold adaptation procedure is provided, the automatic threshold adaptation procedure is operable in one of at least a first and a second adaptation mode, where the first adaptation mode is associated with minimizing queuing delay and adapts the threshold value (L_{th}) on the basis of $n \cdot LC$, where LC represents the estimated link capacity value and $n=1$, and the second adaptation mode is associated with maximizing utilization and adapts said threshold value (L_{th}) on the basis of $m \cdot LC$, where $m>1$ and $m>n$.

In accordance with the present invention, the method of controlling a queue buffer and the queue buffer controller are arranged in such a way that at least two different operation modes are possible, one provided for reducing queuing delay, the other provided for increasing utilization. In the mode for increasing utilization, the threshold value is set higher than in the mode for reducing queuing delay. The reason is that if the threshold is higher, then the average queue length will be longer, but a sender will be able to put more load (e.g. increase the sending window or the sending rate) on the network transporting the data units, because congestion notifications are sent less frequently due to the higher threshold. This enhances utilization, as it means that there will generally always be data units to transport over the link, but also increases queuing delay. On the other hand, if it is desired to reduce queuing delay, then the threshold at which the congestion notification procedure may set in should be lower, thereby making the data unit senders reduce their load on the network more frequently, and thereby reducing the average queuing delay. However, reducing the load also bears a higher risk of reducing the utilization, because the load might get reduced to a degree where there are periods in which there are no

data units to be transported, i.e. in which the link is under-utilized. The present invention therefore provides a queue control method and system that are flexible in the balancing of the trade-off between delay and utilization, by
5 providing respective modes for each situation.

The setting of the first and second mode can be achieved in any desirable way. For example, the setting can be done manually by an operator, e.g. the operator sets a parameter
10 that the control procedure uses to identify which mode is to be used. According to another example, the setting of modes can be done automatically. Preferably, the automatic setting takes into account the number of data unit loss indication events occurring outside of the transmission device in which
15 the buffer being controlled is provided. Namely, in a preferred embodiment, an automatic mode setting procedure sets the threshold adaptation to the second mode (higher threshold) if the number of data units loss indication events exceeds a predetermined value. The exceeding of the
20 predetermined value indicates that data unit losses are occurring, where these data unit losses will lead to the data unit senders reducing their load, such that the additional performance of congestion notifications (e.g. data unit dropping and/or data unit marking) could lead to an
25 unnecessary further load decrease, which in the end would lead to link under-utilization.

Expressed differently, the embodiment just described assumes that the flow control performed at the sender and/or the
30 receiver is such that the sender will reduce its load onto the network transporting the data units (i.e. send less data units) if it determines that a data unit loss has occurred. The purpose of congestion notifications given at the queue buffer being controlled is also to indicate to the sender to
35 reduce the load on the network. As a consequence, the embodiment has the advantage of being able to adjust the issuing of congestion notifications in dependence on

potential data losses outside of the data unit transmission device in which the buffer is provided. This makes it possible to avoid congestion notifications (data unit dropping and/or data unit marking) in cases where data unit losses are occurring anyhow, such that the congestion notifications could lead to the sender reducing its load more than necessary, which in turn could lead to an under-utilization of the link at which the queue buffer is provided. Under-utilization implies a state in which the link is idle, i.e. not transporting any data units. It is very desirable to avoid a link under-utilization, as an idle link is a waste of resources.

In accordance with another preferred embodiment, the method of controlling a queue buffer and the queue buffer controller are arranged in such a way that events which indicate a potential data unit loss in a flow that is being queued in the queue buffer are also taken into account for dynamically adapting the length threshold value within a given adaptation mode. In other words, after a given mode has been set (e.g. either manually or automatically, where the automatic setting may have been done on the basis of a measured number of loss indication events), which means that the threshold value is basically adapted on the basis of $n \cdot LC$ or $m \cdot LC$, where LC represents the link capacity and $m > n$, a dynamic further adjustment can be performed in dependence on the measured number of loss indication events.

It is noted that loss indication events can be any event capable of implying a potential data loss. For example, such an event can be a missing data unit from a sequence of data units being queued (which implies a potential data loss upstream from the queuing buffer) or loss indication information contained in acknowledgment data units sent from the receiver of the queued flow to the sender of the queued flow. Such loss indication information can be explicit, e.g. an explicit notification of the flow receiver to the flow

sender that a particular data unit of the sequence is missing, or implicit e.g. the sending of duplicate acknowledgments for the last data unit received correctly in the sequence.

5

The capability of the automatic threshold adaptation to be operated in the first mode or the second mode implies that a given queue in the buffer being controlled is operated in accordance with one of the modes. It is possible that the
10 buffer holds a plurality of queues, each associated with one of the modes available (there may naturally be more than two operation modes available, e.g. a third mode in which the length threshold is adapted on the basis of $q \cdot LC$, where $q > m$). In other words, each queue has its respective length
15 threshold value for comparing with a respective measured length parameter, and each length threshold is adapted individually. In such a situation, an embodiment is advantageous in which incoming data units that are to be queued, are discriminated into categories associated with the
20 modes, and then placed into a queue operated in accordance with the discriminated mode. For example, the buffer controller can parse the data unit for specific information, e.g. a protocol identifier or port identifier in a header, and assign data units of a delay sensitive type (e.g. data
25 units transporting segments from a Telnet application) to a queue operated in the mode for reducing delay, and assign data units of a throughput sensitive type (e.g. data units transporting segments from an ftp application) to a queue operated in the mode for maximizing utilization.

[Brief description of drawings]

Further aspects and details of the present invention will
5 become apparent from the following detailed description and
preferred embodiments, where reference will be made to the
accompanying figures, in which:

Fig. 1 shows a schematic block diagram representation of a
10 buffer and buffer controller according to the invention,

Fig. 2 shows a flowchart of a method embodiment of the
present invention,

15 Fig. 3 shows a flowchart of another method embodiment of
the present invention, and

Fig. 4 shows a flowchart of a basic method embodiment of
the present invention.

20

[Detailed description of the embodiments]

Although some of the following embodiments may make reference
to specific protocols, such as TCP/IP, the present invention
25 can be applied to any system transporting data units in which
a queue management scheme is used, where a congestion
notification procedure is conducted in dependence on the
event of a queue length parameter reaching a length threshold
value. The present invention is not restricted to any
30 specific such length threshold and congestion notification
scheme, and is therefore e.g. applicable to any known RED
scheme, to schemes that drop data units when a queue is full,
such as tail-drop, random-drop or front-drop, and to any
known scheme that performs explicit congestion notification
35 instead of or in addition to data units dropping.

It is preferable to apply the method and device of the present invention in connection with the active queue management disclosed in EP-A-1249972. The entire disclosure
5 of this document and its US counter-part application are herewith incorporated by reference.

Fig. 1 shows a schematic representation of a queue buffer controller 10 that is capable of implementing the present
10 invention for controlling the management of data units in a queue buffer 20. Reference numeral 3 represents a communication network over which data units 30 arrive at the queue buffer 20, in order to be placed in a queue 21 before being sent over link 40. Reference numeral 50 relates to the
15 data unit transmission device in which the buffer 20 and controller 10 are provided. The data unit transmission device can e.g. be a router or server connected to communication network 3.

20 The data units 30 queued in queue 21 may belong to one or more flows. A flow is generically identified by a source and destination address, the source and destination Service Access Point (SAP) identifier and a protocol identifier. The definition and concept of a flow is well known in the art,
25 e.g. from TCP/IP, in which case the source and destination address are called IP addresses and the SAP identifier is a port address, such that a further explanation is not necessary here.

30 Now specific elements will be described for embodying the concept of the present invention in queue buffer controller 10. It is noted that a queue buffer controller will generally comprise more than these elements, namely known elements for processing received data units and managing the buffer, which
35 are not explicitly described for the sake of simplicity. Especially, the controller 10 may have additional elements

for specifically embodying a system as described in EP-A-1 249 972.

Reference numeral 101 describes a queue length determinator
5 for determining a value of a length parameter related to the
length of queue 21. Furthermore, a comparator 102 is provided
for comparing the determined length value with a length
threshold value Lth provided by a threshold adaptor 104,
which is arranged to automatically adapt the length threshold
10 values by estimating a link capacity value LC based on the
data rate DR of link 40. The comparator 102 is connected to a
congestion notifier 103 that performs a congestion
notification procedure if the determined length value is
greater than the length threshold value. As shall be
15 explained in more detail further on, the length parameter to
be determined can be chosen in any suitable or desirable way,
e.g. be the absolute length queue length QL or an average
queue length QLav , and the congestion notification procedure
can equally be chosen as is suitable or desirable, e.g. be a
20 data unit dropping procedure and/or an explicit data unit
marking procedure.

The congestion notifier 103 preferably comprises a decision
unit 1031 for deciding whether or not to perform a congestion
25 notification with respect to one or more data units in queue
21. In other words, the congestion notifier 103 is arranged
in such a way that it does not necessarily perform a
congestion notification if the queue length parameter exceeds
the length threshold value. This is basically known in the
30 art, e.g. from RED or some systems described in EP-A-1 249
972, where the length threshold value is a first or lower
threshold, and a second or higher length threshold is also
provided, where if the queue length parameter exceeds the
first threshold but does not exceed the second, a probability
35 based decision is made for performing a congestion
notification with respect to one or more data units.

In accordance with the present invention, the threshold adaptor 104 is operable in one of at least two adaptation modes. The first adaptation mode is associated with minimizing queuing delay and adapts the length threshold value L_{th} on the basis of $n \cdot LC$, where $n \geq 1$. For example, n can be 1 and the first adaptation mode sets $L_{th} = LC$ or $L_{th} = LC + \Delta 1$, where $\Delta 1$ is a positive factor smaller than LC , e.g. $0 < \Delta 1 \leq LC/10$. The second adaptation mode is associated with maximizing utilization and adapts the length threshold value L_{th} on the basis of $m \cdot LC$, where $m > 1$ and $m > n$. For example, m can be 3 and the second adaptation mode sets $L_{th} = 3 \cdot LC$ or $L_{th} = 3 \cdot LC - \Delta 2$, where $\Delta 2$ is a positive factor smaller than LC , e.g. $0 < \Delta 2 \leq LC/10$. If the feature of dynamic adaptation with respect to ambient events (such as loss indication events) after mode setting is provided, then the above settings are initial and may be varied thereafter in accordance with the occurrence of such events. If no dynamic adaptation feature is provided, then the value of L_{th} will remain as set above, but it is to be noted that the value of LC is generally not static, i.e. LC will vary over time, and this leads to the possibility of the value of L_{th} changing accordingly.

The factors n and m can be arbitrary positive numbers, but are preferably positive natural numbers.

The setting of the first mode or second mode can be done in any suitable or desirable way. For example, it can be done manually by an operator with the help of an appropriate mechanism, symbolized by switch 106 in Fig. 1, which can be set to the first mode $M1$ or the second mode $M2$. Element 106 may be a real switch, but is preferably a software element with which a user may set the threshold adaptor into a desired mode.

The link capacity value can be understood as the minimal amount of data that the sender of a flow under consideration

must send out, such that the bandwidth that link 40 allocates to said flow is fully used. Full utilization means that the proportion of link bandwidth allocated to the given flow is always in full use. Expressed somewhat differently, if one considers the simplified example of link 40 only serving a single flow, then this means that the sender of that flow sends so much data that link 40 is constantly busy, i.e. constantly sending data units, without any idle time in between. Again in other words, the link capacity is the amount of data that the sender brings into flight, such that any additional data brought into flight will not increase throughput, as the additional data is queued.

The link capacity can therefore also be understood as the product of the data rate provided by the link 40 to the flow in question, multiplied by the round trip time (RTT) associated with said flow for the case of an unloaded network. An unloaded network means that there is no queuing delay. As a consequence, the value of the unloaded RTT is equal to the difference between the actual RTT of the flow and all queuing delays for said flow. The link capacity is sometimes also referred to as the pipe capacity of a hypothetical pipe between the flow end points, said pipe having a "width" DR and a "length" equal to the unloaded RTT.

The estimation of the link capacity value will therefore generally consist in determining a time value indicative of the unloaded RTT, and multiplying this value with the data rate DR provided by link 40 for a flow in question. The determination of this unloaded RTT can be done in any suitable or desirable way. One example is to calculate the sum of a constant RTT_{wc} and the RTT provided by link 40. RTT_{wc} is a worst-case estimation of the overall unloaded RTT excluding the contribution from the link 40, and may have a value of 150 to 300 ms, more preferably 150 to 250 ms. Using this concept has the advantage that no flow specific information needs to be obtained.

An alternative possibility of estimating the unloaded RTT for a given flow consists in calculating the queuing delay at buffer 20, e.g. by keeping an average of the amount of time that a buffered data unit 30 spends in queue 21, and

5 calculating the difference between the actual RTT of the flow and this queuing delay. The value of the actual RTT for the flow can e.g. be inserted into the data units of said flow by the sender and read by controller 10.

10 It is noted that the estimate LC of the link capacity will generally not be identical to the actual momentary link capacity. The process of estimating the link capacity is preferably such that the estimated value exceeds the real link capacity, i.e. the estimate is conservative. This can be
15 achieved in any suitable or desirable way, e.g. by using the above mentioned worst case estimates for the unloaded RTT, and/or by adding predetermined positive factors to one or more of the parameters used in estimating the link capacity. In other words, one can add a predetermined factor to the
20 unloaded RTT and/or to the RTT of the link and/or to DR, and one can add a predetermined positive factor ε to the calculation result, i.e. replacing the calculated value of LC by $LC + \varepsilon$: $LC \leftarrow LC + \varepsilon$.

25 The choice of parameters n and m for the first and second mode is preferably done in accordance with the reaction that a data unit sender will show when receiving a congestion notification. If the sender is of a type that reduces its load on the network by a factor k, e.g. divides its send
30 window by k if window-based flow control is used, then n is preferably chosen to be k-1 and m is preferably chosen to be k^2-1 . The reason will be explained in the following.

As already mentioned above, the estimated link capacity LC is
35 such that once more data units are in flight than this value, queuing begins. As a consequence, when the length threshold value Lth is reached, the amount of data in flight is LC +

Lth. As a consequence, when congestion notifications are started, a window-based sender will have a send window equal to $LC + Lth$.

5 As a condition for setting Lth in a mode that serves to minimize queuing delay, it is desired to achieve a fair balance between holding the average queue length short and keeping the link busy, i.e. it is desired to make the queue length as short as possible without causing under-
 10 utilization. This leads to the consideration that upon onset of congestion notification, i.e. when the send window of $Lth+LC$ is divided by k , the resulting window size be equal to LC , as this means that the load is sufficient to keep the link busy: $\frac{L_{th} + LC}{k} = LC$, which leads to $Lth = LC \cdot (k-1)$.

15

As a condition for setting Lth in a mode that serves to maximize utilization, it is considered that besides the performance of congestion notifications for reducing the load a sender places on the network, data loss indication events
 20 occur, which also lead to the sender reducing its load. Then the following worst case assumption is made: after a first reduction of the send window by k , a data loss event (not an intentional data drop in the context of congestion notification) occurs, provoking a further reduction by k . In
 25 such a scenario that implies the occurrence of data losses outside of the transmission device in which the buffer under control is located, this leads to the condition: $\frac{L_{th} + LC}{k^2} = LC$, which in turn leads to $Lth = LC \cdot (k^2 - 1)$.

30 In TCP/IP $k=2$, i.e. a sender reduces the send window to one half when receiving a congestion notification or determining a data unit loss indication event. As a consequence, if the buffer being controlled is used to queue TCP segments, then n is preferably 1 and m is preferably 3.

35

In accordance with a preferred embodiment of the present invention, a loss indication event detector 105 is additionally provided. The detector 105 is arranged for detecting an event outside of said data unit transmission
5 device 50, which event indicates a potential data unit loss in a flow queued in queue 21.

In the example of Fig. 1, the loss indication event detector 105 is connected to both the threshold adapter 104 and the
10 decision unit 1031, such that both can take detected loss indication events or signals derived therefrom into account. However, it is also within the scope of the present embodiment that only the threshold adapter 104 take the results of the loss indication event detector 105 into
15 account, or that only the decision unit 1031 take the results of the loss indication event detector 105 into account.

Further details of procedures conducted by the above described elements will be described later in connection with
20 the method examples of the invention that can be embodied in the buffer controller of Fig. 1.

It is noted that the above-described elements 101-105 can be provided as hardware, software or any suitable combination of
25 hardware and software. Preferably, the controller 10 is a programmable data processor, and the elements 101 to 105 are software elements, e.g. program code parts.

Fig. 4 shows a flow chart of a basic embodiment of the method
30 of the present invention, which method can be performed with the controller 10 shown in Fig. 1. In first step S1, a value of a length parameter related to the length of the queue 21 is determined. This queue length related parameter can be related to the queue length in any desirable or suitable way,
35 e.g. can be the actual or momentary queue length QL, or a parameter derived from the actual or momentary queue length, such as an average value QLav.

In the example of Fig. 4, the queue length related parameter is the actual queue length QL. If it is desirable to use an average queue length QLav, this average value can be

5 determined in accordance with any known suitable averaging algorithm, and such an algorithm may typically consist in updating an old average value by calculating the sum of the old average multiplied by a first weight factor and the momentary queue length multiplied by a second weight factor.

10 For example, QLav can be calculated as

$$\text{QLav (new)} = \text{QLav (old)} \times (1 - 1/2^{\text{wf}}) + (\text{QL} \times 1/2^{\text{wf}})$$

where QL represents the momentary queue length value and wf is an exponential weight factor adjustable between 0 and 1.

15 Returning to Fig. 4, in step S2 the queue length parameter QL is compared with a length threshold value Lth. If the length threshold value Lth is exceeded, then a congestion notification procedure S3 is performed, otherwise the
20 congestion notification procedure S3 is skipped.

As already mentioned above, the congestion notification procedure can be chosen in any suitable or desirable way. For example, it can comprise dropping/marking one or more
25 predetermined or randomly selected data units from the queue 21, or dropping/marking one or more newly arrived data units before placing them into queue 21.

The congestion notification procedure may also comprise a
30 decision procedure for deciding whether or not to actually perform a congestion notification with respect to one or more of the data units from queue 21. As already mentioned above, such a decision procedure can e.g. depend on a probability function, as is known in the prior art.

35 In the example of Fig. 4, the control procedure then continues to step S5, in which the automatic threshold

adaptation procedure is conducted. As already specified above, the automatic threshold adaptation procedure S5 is operable in one of at least a first and a second adaptation mode, the first adaptation mode being associated with
5 minimizing queuing delay and adapting the length threshold value L_{th} on the basis of $n \cdot LC$, where $n=1$, and the second adaptation mode being associated with maximizing utilization and adapting the length threshold value L_{th} on the basis of $m \cdot LC$, where $m>1$ and $m>n$.

10 The mode in which S5 operates can be set manually by an operator, or can be set automatically by a routine dedicated to this task. An example for such a routine will now be described.

15 Fig. 2 shows a method embodiment that comprises all of the steps of the embodiment of Fig. 4, such that a repeated description is not necessary, and where a loss indication event detection procedure S4 is added after steps S2, S3.

20 This loss indication event detection procedure S4 is capable of detecting events that indicate a potential data unit loss in one or more of the flows queued in queue 21, where this potential data unit loss occurs outside of data unit transmission device 15. For example, the loss indication
25 event detection procedure S4 can comprise monitoring sequence identifiers of data units of a queued flow, where the missing of a data unit from the sequence indicates a potential data unit loss. Alternatively or additionally, the loss indication event detection procedure may comprise monitoring loss
30 indication information in acknowledgment data units sent from the flow receiver to the flow sender, where this loss indication information can be explicit or implicit.

In accordance with the example of Fig. 2, the automatic
35 threshold adaptation procedure S5 takes into account results provided by the loss indication event detection procedure S4, in order to automatically set the first or second adaptation

mode. For example, the loss indication event detection procedure S4 can output a count value indicative of the number of such loss indication events (which will be explained in more detail further on), and if this count value exceeds a predetermined value then the adaptation procedure is set to operate in the second mode (utilization optimisation), otherwise it is set in the first mode (delay optimisation).

Beyond using the outcome of the loss indication event detection procedure for automatically setting the adaptation mode, it is additionally or alternatively also possible to use the outcome of the loss indication event detection procedure for dynamically adapting the threshold value L_{th} in the first or second mode. In other words, after a given mode has been set (e.g. either manually or automatically, where the automatic setting may have been done on the basis of a measured number of loss indication events), which means that the threshold value is basically adapted on the basis of $n \times LC$ or $m \times LC$, a dynamic further adjustment can be performed in dependence on the measured number of loss indication events.

For example, if the loss indication event detection procedure indicates a certain amount of potential data units losses outside of the transmission device 50, then the threshold L_{th} used for triggering the congestion notification procedure can be increased beyond the value that it was set to in the given mode. For example, it can be increased beyond the estimated link capacity value LC , which could be an initial choice for L_{th} in the first mode. The increase will be only a fraction of LC , e.g. up to one tenth of LC , as a sort of "fine tuning" within a given mode.

The increasing of the threshold value L_{th} has the effect of making it less probable that the queue length parameter QL will reach the threshold L_{th} in step S2, such that it is less probable that a congestion notification will be performed.

Thereby, it can be avoided that congestion notifications are performed although the flow sender is already reducing the load onto the network due to data unit losses that occur independently of the congestion notification procedure. Data unit losses outside of the transmission unit 50 together with congestion notifications from the transmission unit 50 could lead to the flow sender reducing its load onto the network too much, which in turn could lead to an under-utilization of link 40.

Attention is drawn to the fact that the specific arrangement of steps shown in Fig. 2 and 4 is only an example. Especially, steps S1, S2 and S3, which together form a procedure for deciding on the triggering of a congestion notification procedure, are independent of the adaptation procedure embodied by step S5. Consequently, steps S1-S3 may be arranged independently of S5. Equally, step S4 can be arranged independently of S1-S3 and S5, e.g. as a procedure that runs in parallel to the others and/or is selectively invoked as a sub-routine.

Regarding the examples of Fig. 2 and 4, it is also noted that all of the shown steps will generally be contained in a larger method of controlling or managing the queue buffer 20, which larger method has more steps and procedures, but where these additional steps and procedures are not shown as they do not pertain to the present invention. The methods of Fig. 2 and 4, just as all method embodiments of the invention, may be implemented as software, where steps S1-S3 can e.g. be implemented in one thread, while S5 can be implemented in another independent thread. Step S4 can be implemented in yet another independent thread.

In the example of Fig. 2 the adaptation procedure for Lth in step S5 can take the outcome of the loss indication event detection procedure S4 into account in any desired or suitable way. In general, the threshold adaptation procedure

will be arranged in such a way that the length threshold L_{th} will be increased with increasing occurrence of potential data losses outside of the data unit transmission device 50, in order to make the triggering of the congestion

5 notification procedure S3 less likely, as explained above.

In accordance with a preferred embodiment, the method of loss event detection is arranged in such a way that a counting procedure is provided for counting the number of data unit
10 loss indication events occurring outside of transmission device 50 in the queued flow under consideration, and a procedure is provided for deriving a characteristic count value from that counted numbers. The automatic threshold adaptation procedure then comprises a step for setting the
15 adaptation mode and/or fine tuning the length threshold value L_{th} in dependence on this characteristic count value.

The process of deriving a characteristic count value can be chosen in any suitable or desirable way. For example, it may
20 comprise determining the number of loss indication events occurring outside of data unit transmission device 50 in the queued flow under consideration in each of p respective predetermined time intervals, where p is a natural number, and then selecting a maximum among that numbers as the
25 characteristic count value. If p is 1, then this means simply counting the number of loss indication events within a predetermined interval, and outputting the count value for each consecutive interval. In order to make the characteristic count value less susceptible to short-time
30 fluctuations, it is preferable to choose p larger than 1, e.g. $p = 5$. In this case the procedure holds the counted number of loss indication events in the five latest intervals, and outputs the maximum among the loss indication events of the latest five intervals as the characteristic
35 count value.

The predetermined interval or intervals during which the number of loss indication events are counted can be chosen in any suitable or desirable way, e.g. have a predetermined fixed length. Preferably, they are defined dynamically as the time between two consecutive decisions of performing congestion notification for one or more data units by the decision procedure contained in the congestion notification procedure. In other words, once the decision procedure in the congestion notification procedure decides to perform a congestion notification, a new interval is started and consequently a new count of loss indication events begins. This count of loss indication events is continued until the next performance of a congestion notification.

Another possibility of deriving a characteristic count value consists in determining an average number of loss indication events occurring outside of the transmission device 50. Such an averaging can be conducted by monitoring the number of loss indication events in consecutive predetermined intervals, just like in the example described above, and then averaging the thus counted numbers over the number of intervals. Such an averaging operation is preferably conducted as a running average, e.g.

$$AV(\text{new}) = AV(\text{old}) \times (1-q) + \text{Num} \times q,$$

where AV represents the average value, q is a weighting factor $0 < q < 1$, and Num is the number of loss indication events in the latest complete interval.

As already mentioned in general, the length threshold adaptation procedure S5 will increase the length threshold value Lth if the characteristic count value indicates an increase in loss indication events.

Regarding the counting of loss indication events, it is noted that the choice of what is counted as a loss indication event

can be performed in any suitable or desirable way. For example, if the loss indication event detection procedure comprises monitoring the sequence identifiers of data units in the flow under consideration, then each detection of a missing data unit in the sequence ("a gap" in the sequence) can be counted as a loss indication event. This process can also be modified in that the recognition of such a gap is not immediately counted as a loss indication event, but that the procedure first waits for a predetermined wait period, and a loss indication event is only counted if the missing data unit does not appear within the wait period. The introduction of such a wait period makes the procedure more resistant against reordering of data units over the transmission path in network 3.

As another example, if the loss indication event detection procedure comprises monitoring loss indication information in acknowledgment data units, then e.g. each occurrence of a loss indication information can be counted as a loss indication event. Taking TCP/IP as an example, a loss indication event can be the occurrence of a duplicate acknowledgment. As such, each occurrence of a duplicate acknowledgment can be counted as a loss indication event. However, when monitoring acknowledgment data units for loss indication information, is preferable to have the buffer controller operate in the same way as the data unit sender with respect to the identification of data unit loss events. In other words, the buffer controller should apply the same criteria for judging a data unit loss event as the data unit sender for the purpose of flow control. For example, in TCP/IP a data unit sender will judge that a data unit loss has occurred after having received a predetermined number (e.g. 3) of duplicate acknowledgments. In this case, it is preferable that the buffer controller also counts the duplicate acknowledgments, and judges a data unit loss event to have occurred in the predetermined number of duplicate acknowledgments has been counted.

If the congestion notification performed by the buffer controller are data unit drops, then the controller should preferably be arranged to record these drops in association with the sequence number of the dropped data unit, in order to be able to identify those acknowledgment messages that relate to said dropped data unit, and to not count such loss indication information as an event that indicates a data loss outside of the transmitting device 50.

Now a further embodiment of the present invention will be described with reference to Fig. 3. Steps S1 and S2 are identical to what has already been described in connection with Fig. 2 and 4, such that a repeated description is not necessary. However, in the example of Fig. 3, the congestion notification procedure S3 consists of steps S31, S32 and S33, which are different from the example of Fig. 2 or 4. Step S31 represents a decision procedure for deciding whether to perform a congestion notification with respect to one or more data units. For example, this can be the above mentioned example of examining a probability function. If the outcome of this decision procedure is affirmative, then step S32 is conducted, i.e. a procedure for determining whether an event indicating a potential data unit loss outside of the data transmission device 50 has occurred for the flow under consideration, and if such an event has occurred, the decision of performing a congestion notification in step S31 is cancelled. In other words, in this case no congestion notification is performed.

In the specific example of Fig. 3, step S32 consists in determining whether a loss indication event occurs within a predetermined guard time GT after it is detected that the value of the length parameter QL has exceeded the length threshold value Lth.

In other words, after the detection of QL exceeding Lth, it is monitored whether a loss indication event LIE occurs within the guard time GT or not. If a loss indication event occurs, then the congestion notification decided in step S31 is cancelled or disabled, i.e. not performed, because it is assumed that the flow sender will reduce its load into the network based upon the indicated data unit loss outside of the transmission device 50, such that the performance of the congestion notification could lead to an excessive reduction of sent data units on the part of the flow sender, thereby possibly leading to an under-utilization of link 40.

On the other hand, if within the procedure of step S32 no loss indication event is detected within the guard time GT, the congestion notification decided in step S31 is performed in step S33.

In the above example of Fig. 3, the decision procedure, which consists of steps S31, S32 and S33 takes the loss indication event detection outcome into account. In the example of Fig. 2, it was the automatic threshold adaptation procedure S5 that took the loss indication event detection outcome into account. It is noted that these two possibilities of making use of the loss indication event detection outcome can also be combined, i.e. that both the automatic threshold adaptation procedure and the decision procedure take the outcome of the loss indication event detection into account.

Now a further embodiment will be described. The capability of the automatic threshold adaptation S5 or threshold adaptor 104 to be operated in the first mode or the second mode implies that a given queue in the buffer being controlled is operated in accordance with one of the modes. It is possible that the buffer holds a plurality of queues, each associated with one of the modes available. There may naturally be more than two operation modes available, e.g. a third mode in which the length threshold is adapted on the basis of $s \times LC$,

where $s > m$. In other words, each queue has its respective length threshold value for comparing with a respective measured length parameter, and each length threshold is adapted individually. In such a situation, an embodiment is proposed in which incoming data units that are to be queued, are discriminated into categories associated with the modes, and then placed into a queue operated in accordance with the discriminated mode. For example, the buffer controller can parse the data unit for specific information, e.g. a protocol identifier or port identifier in a header, and assign data units of a delay sensitive type (e.g. data units transporting segments from a Telnet application) to a queue operated in the mode for reducing delay, and assign data units of a throughput sensitive type (e.g. data units transporting segments from an ftp application) to a queue operated in the mode for maximizing utilization.

Although the present invention has been described by way of specific examples, these are not intended to be limiting, as the scope of the invention is defined by the appended claims. Reference signs in the claims only serve to make the claims easier to read and are also not intended to have any limiting effect.